Lớp JFileChooser trong Java Swing

Lớp **JFileChooser** là một thành phần cung cấp một kỹ thuật đơn giản cho người dùng để lựa chọn một file. Cú pháp khai báo cho lớp Javax.swing.JFileChooser là:

```
public class JFileChooser
   extends JComponent
      implements Accessible
```

Lớp này kế thừa các phương thức từ các lớp sau:

- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

Các constructor của lớp JFileChooser trong Java Swing

1. JFileChooser(): Xây dựng một JFileChooser trỏ tới thư mục mặc định của người dùng.

2. JFileChooser(File currentDirectory): Xây dựng một JFileChooser bởi sử dụng File đã cho như là path.

3. JFileChooser(File currentDirectory, FileSystemView fsv): Xây dựng một JFileChooser bởi sử dụng thư mục hiện tại đã cung cấp và FileSystemView.

4. JFileChooser(FileSystemView fsv): Xây dựng một JFileChooser bởi sử dụng FileSystemView đã cho.

5. JFileChooser(String currentDirectoryPath): Xây dựng một JFileChooser bởi sử dụng path đã cho.

6. JFileChooser(String currentDirectoryPath, FileSystemView fsv): Xây dựng một JFileChooser bởi sử dụng thư mục hiện tại đã cung cấp và FileSystemView.

Các phương thức của lớp JFileChooser trong Java Swing

STT	Phương thức & Miêu tả
1	**void removeActionListener(ActionListener l)** Gỡ bỏ một ActionListener từ file chooser
2	**boolean removeChoosableFileFilter(FileFilter f)** Xóa một trình lọc filter từ danh sách các trình lọc file có thể lựa chọn của người dùng
3	**void rescanCurrentDirectory()** Nói cho UI rằng cần quét lại danh sách file của nó từ thư mục hiện tại
4	**void resetChoosableFileFilters()** Phục hồi danh sách trình lọc file có thể lựa chọn về trạng thái ban đầu của nó
5	**void setAcceptAllFileFilterUsed(boolean b)** Xác định xem AcceptAll FileFilter có được sử dụng như là một lựa chọn có sẵn không trong danh sách trình lọc có thể chọn
6	**void setAccessory(JComponent newAccessory)** Thiết lập thành phần phụ thêm
7	**void setApproveButtonMnemonic(char mnemonic)** Thiết lập mnemonic của nút xác nhận bởi sử dụng một ký tự
8	**void setApproveButtonMnemonic(int mnemonic)** Thiết lập mnemonic của nút xác nhận bởi sử dụng một keycode dạng số

9	**void setApproveButtonText(String approveButtonText)** Thiết lập text được sử dụng trong ApproveButton trong FileChooserUI
10	**void setApproveButtonToolTipText(String toolTipText)** Thiết lập tooltip text được sử dụng trong ApproveButton
11	**void setControlButtonsAreShown(boolean b)** Thiết lập thuộc tính chỉ rằng có hay không các nút xác nhận và hủy bỏ được hiển thị trong file chooser
12	**void setCurrentDirectory(File dir)** Thiết lập thư mục hiện tại
13	**void setDialogTitle(String dialogTitle)** Thiết lập chuỗi trong thanh tiêu đề của cửa sổ JFileChooser
14	**void setDialogType(int dialogType)** Thiết lập kiểu của dialog này
15	**void setDragEnabled(boolean b)** Thiết lập thuộc tính dragEnabled property, mà phải là true để kích hoạt trình xử lý hoạt động drag tự động (phần đầu tiên của hoạt động drag và drop) trên thành phần này
16	**void setFileFilter(FileFilter filter)** Thiết lập trình lọc file hiện tại
17	**void setFileHidingEnabled(boolean b)** Bật hoặc tắt chế độ ẩn file

18	**void setFileSelectionMode(int mode)** Thiết lập JFileChooser để cho phép người dùng: chỉ lựa chọn file, chỉ thư mục hoặc cả hai
19	**void setFileSystemView(FileSystemView fsv)** Thiết lập file system view mà JFileChooser sử dụng để truy cập và tạo các nguồn file system, chẳng hạn như tìm đĩa mềm và lấy danh sách các root drive
20	**void setFileView(FileView fileView)** Thiết lập file view để được sử dụng để lấy thông tin UI, chẳng hạn như icon mà biểu diễn một file hoặc miêu tả kiểu của file
21	**void setMultiSelectionEnabled(boolean b)** Thiết lập file chooser để cho phép lựa chọn multi-file
22	**void setSelectedFile(File file)** Thiết lập file được lựa chọn
23	**void setSelectedFiles(File[] selectedFiles)** Thiết lập danh sách các file được lựa chọn nếu file chooser được thiết lập dạng multi-file
24	**protected void setup(FileSystemView view)** Thực hiện khởi tạo và thiết lập constructor chung
25	**int showDialog(Component parent, String approveButtonText)** Popup một hộp thoại file chooser tùy biến với một nút xác nhận tùy biến
26	**int showOpenDialog(Component parent)**

	Hiển thị một hộp thoại "Open File"
27	**int showSaveDialog(Component parent)** Hiển thị một hộp thoại "Save File"
28	**void updateUI()** Phục hồi thuộc tính UI về giá trị của L&F hiện tại
29	**boolean accept(File f)** Trả về true nếu file nên được hiển thị
30	**void addActionListener(ActionListener l)** Thêm một ActionListener tới file chooser
31	**void addChoosableFileFilter(FileFilter filter)** Thêm một bộ lọc filter tới danh sách bộ lọc các file có thể lựa chọn của người dùng
32	**void approveSelection()** Được gọi bởi UI khi người dùng nhấn nút xác nhận (được gán nhãn "Open" hoặc "Save", theo mặc định)
33	**void cancelSelection()** Được gọi bởi UI khi người dùng chọn nút Cancel
34	**void changeToParentDirectory()** Các thay đổi tới thư mục để được thiết lập tới thư mục cha của thư mục hiện tại
35	**protected JDialog createDialog(Component parent)** Tạo và trả về một JDialog mới đang bao hộp thoại này được căn chỉnh vào giữa frame

	của thành phần cha
36	**void ensureFileIsVisible(File f)** Đảm bảo rằng file đã cho là nhìn thất, không bị ẩn
37	**protected void fireActionPerformed(String command)** Thông báo cho tất cả Listener mà đã đăng ký nhận thông báo trên kiểu sự kiện (event type) này
38	**FileFilter getAcceptAllFileFilter()** Trả về trình lọc file là AcceptAll

Chương trình ví dụ lớp JFileChooser

```
package com.vietjack.gui;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SwingControlDemo {

   private JFrame mainFrame;
   private JLabel headerLabel;
   private JLabel statusLabel;
   private JPanel controlPanel;

   public SwingControlDemo(){
      prepareGUI();
   }

   public static void main(String[] args){
```

```
      SwingControlDemo  swingControlDemo = new SwingControlDemo();
      swingControlDemo.showFileChooserDemo();
   }

   private void prepareGUI(){
      mainFrame = new JFrame("Vi du Java Swing");
      mainFrame.setSize(400,400);
      mainFrame.setLayout(new GridLayout(3, 1));
      mainFrame.addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
         }
      });
      headerLabel = new JLabel("", JLabel.CENTER);
      statusLabel = new JLabel("",JLabel.CENTER);

      statusLabel.setSize(350,100);

      controlPanel = new JPanel();
      controlPanel.setLayout(new FlowLayout());

      mainFrame.add(headerLabel);
      mainFrame.add(controlPanel);
      mainFrame.add(statusLabel);
      mainFrame.setVisible(true);
   }

   private void showFileChooserDemo(){
      headerLabel.setText("Control in action: JFileChooser");

      final JFileChooser  fileDialog = new JFileChooser();
      JButton showFileDialogButton = new JButton("Open File");
```

```
      showFileDialogButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            int returnVal = fileDialog.showOpenDialog(mainFrame);
            if (returnVal == JFileChooser.APPROVE_OPTION) {
               java.io.File file = fileDialog.getSelectedFile();
               statusLabel.setText("File Selected :"
               + file.getName());
            }
            else{
               statusLabel.setText("Open command cancelled by user." );
            }
         }
      });
      controlPanel.add(showFileDialogButton);
      mainFrame.setVisible(true);
   }
}
```